IMAGE COMPRESSION VIA INDEPENDENT COMPONENT ANALYSIS

Mónica F. Bugallo, Adriana Dapena, Luis Castedo

Abstract—**Independent Component Analysis (ICA) is an extension of the well-known technique for multivariate data analysis called Principal Component Analysis (PCA). The main difference between both techniques is that ICA provides independent components instead of uncorrelated components. As a consequence, ICA is more adequate for applications such as image compression, where the most important information is contained in the high-order relationships among the pixels. In this paper, we propose an ICA-based algorithm for image compression which is an extension of the EASI algorithm (Equivariant Adaptive Separation via Independence) proposed by Cardoso and Laheld [6]. We also compare the performance of our algorithm with PCA and with the Discrete Cosine Transform (DCT).**

Keywords—**Independent component analysis, blind source separation, self-organized learning, image compression**

This work has been supported by FEDER (grant 1FD97-0082).

I. Introduction

Self-organized neural networks have been widely used for extracting important characteristics in a data set [1]. For instance, the learning algorithms based on Principal Component Analysis (PCA) extract uncorrelated components (principal components). In general, we can obtain a substantial dimensionality reduction because few principal components contain most of the intrinsic information in the input [1]. However, PCA-based algorithms do not consider the information contained in the high-order correlations.

Independent component analysis (ICA) is a generalization of PCA which incorporates high-order moments [2]. The aim in ICA is to extract statistically independent components from the input data instead of uncorrelated components. As a consequence, ICA looks more adequate than PCA for applications where there exists high-order relationships in the input data, such as image compression [3] and face recognition [4]. In fact, our simulation results show that few statistically independent components contain most information of the input data and these components preserve more information than the principal components.

This paper is structured as follows. Section II reviews some concepts about ICA and explains the relationship between ICA and image compression. In Section III we propose an image compression algorithm which is based on the EASI (Equivariant Adaptive Separation via Independence) algorithm proposed by Cardoso and Laheld [6]. In Section IV we analyze the stability of our algorithm. Section V presents several simulation results. Finally, Section VI is devoted to the conclusions.

II. Independent Component Analysis

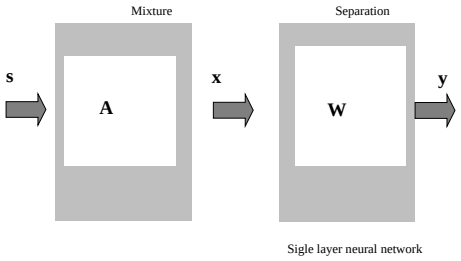

Fig. 1. Blind source separation model.

A number of ICA-based algorithms have been developed in the context of Blind Source Separation (BSS) [5], [6]. The basic model in ICA and BSS is shown in Figure 1. It is assumed that a set of zero-mean statistically independent non-Gaussian distributed signals (sources), $\mathbf{s} = [s_1(n), \cdots, s_N(n)]^T$, are combined using a $N \times N$ non-singular matrix, $\mathbf{A}$, to obtain a vector of observations $\mathbf{x} = [x_1(n), \cdots, x_N(n)]^T$ given by

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1}$$

The objective in BSS is to recover all the components in $\mathbf{s}$ from the observations $\mathbf{x}$ without knowing the sources nor the mixing system. Towards this aim, the observations are processed by a linear neural network with N^2 synaptic weights w_{ij} and N outputs, $y_i(n) = \sum_{j=1}^{N} w_{ij} x_j(n)$. In a compact form, we can write

$$\mathbf{y} = \mathbf{W}^T\mathbf{x} \tag{2}$$

Departamento de Electrónica e Sistemas. Universidade da Coruña. Campus de Elviña s/n, 15.071. A Coruña. Tel: ++34-981-167150, Fax: ++34-981-167160, e-mail: adriana@des.fi.udc.es.

where $\mathbf{y}$ is a $N \times 1$ vector and $\mathbf{W}$ is a $N \times N$ matrix containing the synaptic weights w_{ij}.

A way to recover the sources is to find a matrix $\mathbf{W}$ such as the outputs are statistically independent. The justification of this idea is a consequence of the Darmois-Skitovich theorem [7], [8]: if $\mathbf{s}$ is a vector of statistically independent signals with non-Gaussian distribution then $\mathbf{y} = \mathbf{Gs} = \mathbf{W}^T\mathbf{As}$ is a vector of statistically independent signals if and only if $\mathbf{G} = \mathbf{\Delta P}$. This means that the statistical independence of the output implies source separation.

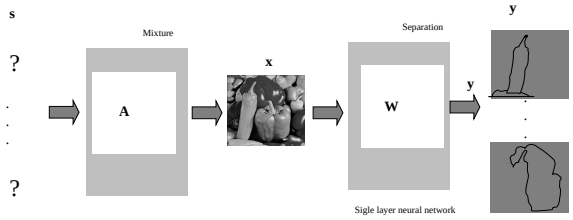

Fig. 2. Image synthesis model.

It is interesting to note the relationship between the BSS problem and image compression [9]. Figure 2 shows the image synthesis model. We can interpret $\mathbf{x}$ as a linear synthesis of the unknown components in $\mathbf{s}$. The goal in image compression is to obtain a basis matrix $\mathbf{W}$ so that $\mathbf{y} = \mathbf{W}^T\mathbf{x}$ are the underlying components. Compression is achieved because we do not retain and code all the components in $\mathbf{y}$ but only a number $p < N$ (typically those that contain larger energy [10]). Let us denote $\hat{\mathbf{y}}$ to the $N \times 1$ vector that contains the retained components. The recovered image can be computed as

$$\hat{\mathbf{x}} = \mathbf{W}^{-T}\hat{\mathbf{y}} \tag{3}$$

where $\mathbf{W}^{-T}$ denotes the inverse of $\mathbf{W}^T$.

III. Generalized EASI Algorithm

In this section we propose an image compression algorithm called Generalized Equivariant Adaptive Separation via Independence (GEASI) which is based on the algorithm proposed by Cardoso and Laheld in the context of BSS [11]. The encoding algorithm works as follows. First, the image is partitioned into blocks of size $n \times n$, $\mathbf{X}$, and then these blocks are converted into $n^2 \times 1$ vectors, $\mathbf{x}$. The weight matrix, $\mathbf{W}$, is computed using the learning rule

$$\begin{aligned}\mathbf{W}(k+1) &= \mathbf{W}(k) - \mu\mathbf{W}(k)(E[\mathbf{y}\mathbf{y}^T] - \mathbf{I} \\ &+ E[\mathbf{y}\mathbf{g}^T(\mathbf{y})]\mathcal{E} - \mathcal{E}E[\mathbf{g}(\mathbf{y})\mathbf{y}^T])\end{aligned} \tag{4}$$

where $\mathbf{g}^T(\mathbf{y}) = [g(y_1), ..., g(y_{n^2})]^T$ contains a nonlinear function $g(.)$, $\mathbf{I}$ is a $n^2 \times n^2$ identity matrix and $\mathcal{E}$ is a $n^2 \times n^2$ diagonal matrix containing ± 1. In practice, the statistical moments in (4) are estimated using L samples of the outputs and each observation $\mathbf{x}$ is passed P times.

Note that the first term in (4), $E[\mathbf{y}\mathbf{y}^T] - \mathbf{I}$, extracts uncorrelated components, $E[\mathbf{y}\mathbf{y}^T] = \mathbf{I}$. However, the nonlinear function $g(.)$ incorporates high-order statistics to the learning rule and, therefore, the GEASI algorithm extracts independent components. The main difference between the learning rule (4) and the EASI algorithm [11] is that we have included a diagonal matrix $\mathcal{E}$ because we have no information about the statistical parameters of the components. Recall that we only suppose that they are statistically independent and have zero-mean. The stability analysis in Section IV gives some clues to find the matrix $\mathcal{E}$.

Note that both the observation vector and the output vector have the same dimension, n^2. In order to obtain a dimensionality reduction, we need to determine which outputs must be retained in the encoder. In theory, we must retain the components closer to the original image. However, this criterion is computational expensive because we must compute the mean square error between the original image and the image obtained from each component. For this reason, we propose to retain the p components with largest energy, $E[y_i^2]$.

The decoding process consists of evaluating the expression (3) using the retained components. Note that the decoder also need to know the weight matrix $\mathbf{W}$. However, this overhead is small in comparison with the image size.

IV. Stability Analysis of the GEASI Algorithm

The stability of GEASI algorithm can be analyzed using concepts of dynamic systems [1], [12]. We will suppose that μ is a small positive quantity and $\mathbf{x}$ is stationary. Then, the stationary points of the recursion (4) corresponds to the roots of the Ordinary Differential Equation given by

$$\begin{aligned}\frac{d\mathbf{W}}{dt} &= -\mathbf{W}(E[\mathbf{y}\mathbf{y}^T] - \mathbf{I} \\ &+ E[\mathbf{y}\mathbf{g}^T(\mathbf{y})]\mathcal{E} - \mathcal{E}E[\mathbf{g}(\mathbf{y})\mathbf{y}^T])\end{aligned} \tag{5}$$

Note that the solutions where statistically independent components are extracted, $\mathbf{y} = \mathbf{s}$, $E[s_i^2] = 1$, $i = 1, ..., n^2$ are roots of (5). In addition, in appendix A we demonstrate that these points are attractors of (4) when the following condition is satisfied

$$\mathcal{C}_{s_i} + \mathcal{C}_{s_j} < 0 \quad i, j = 1, ..., n^2, \; i \neq j \tag{6}$$

where $\mathcal{C}_{s_i} = \epsilon_i(E[s_i g(s_i)] - E[g'(s_i)])$, ϵ_i is the $i-$th entry of $\mathcal{E}$ and $g'(.)$ is the first derivative of $g(.)$. From the stability analysis, we can devise two methods for choosing the matrix $\mathcal{E}$ in order to extract the components in the original image without knowing their statistical characteristics. The simplest method consists of using a matrix fixed *a priori*: $\epsilon_i = 1$, $i = 1, ..., l$ and $\epsilon_i = -1$, $i = l+1, ..., n^2$. In this case, we suppose that

$$\begin{aligned}C_{s_i} = E[s_i g(s_i)] - E[g'(s_i)] < 0, \; i = 1, ..., l \\ C_{s_i} = E[s_i g(s_i)] - E[g'(s_i)] > 0, \; i = l+1, ..., n^2\end{aligned} \tag{7}$$

The second method consists of determining the parameters ϵ_i in each iteration in order to obtain $C_{y_i} < 0,\ i = 1, ..., n^2$, i.e., we evaluate the expression

$$C_{y_i} = E[y_i g(y_i)] - E[g'(y_i)],\ i = 1, ..., n^2 \qquad (8)$$

where the expectations are computed using L samples of the output y_i. Then, we set $\epsilon_i = -1$ when $C_{y_i} > 0$ and $\epsilon_i = 1$ when $C_{y_i} < 0$.

V. Simulation results

In this section the results of several computer simulations are presented to illustrate the performance of the GEASI algorithm. For comparison, we also present the results obtained using the decorrelating algorithm

$$\mathbf{W}(k+1) = \mathbf{W}(k) - \mu \mathbf{W}(k) \left(E[\mathbf{y}\mathbf{y}^T] - \mathbf{I} \right) \qquad (9)$$

and retaining the outputs with most energy. We have used a step-size $\mu = 10^{-4}$, estimation length $L = 20$ and $P = 20$ passes of the data. We have also compared our results with the one-dimensional Discrete Cosine Transform (DCT) [13]. The performance has been measured using the peak signal to noise rate (PSNR)

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}\ \ dB \qquad (10)$$

where MSE is the mean square error between the original and the recovered image.

In the first experiment, we have considered the 512×512 image in Figure 3 which has been partitioned into 4×4 blocks. Each block has been separately made zero-mean. The separating system has processed 16,384 observations $\mathbf{x}$ of dimension 16×1 randomly selected. The 16×16 basis matrix $\mathbf{W}$ was learned using the algorithm (4) with a fixed step-size parameter $\mu = 10^{-4}$, estimation length $L = 20$, $P = 20$ passes of the data, the non-linear function $g(y_i) = tanh(y_i)$, $\epsilon_i = -1,\ i = 1, ..., 8$ and $\epsilon_i = 1,\ i = 9, ..., 16$. Figure 4 (left) shows the images obtained from the outputs with largest energy. It is apparent that the images have high visual quality although the image obtained from only one component presents a block effect. For comparison, Figure 4 (right) shows the images obtained using the decorrelating algorithm (9) and retaining several principal components. Figure 5 shows the PSNR obtained using the proposed algorithm, the decorrelating algorithm (9) and the DCT. We can see that the GEASI algorithm provides the best results.

Recall that in the GEASI algorithm, we retain the components with largest energy. In order to validate this decision criterion, Table I presents the energy of each output and the MSE between the original image and the image obtained using the corresponding component. We can see that the outputs with largest energy have the minimum MSE. On the other hand, small energy implies larger MSE.

We have also tested the behavior of the GEASI algorithm for several non-linear functions. Figure 6 shows the PSNR obtained using the sigmoid function and the hyperbolic tangent function. The matrix $\mathcal{E}$ is estimated in each iteration (aprox. I) or it is a fixed matrix with $\epsilon_i = -1,\ i = 1, ..., 8$ and $\epsilon_i = 1,\ i = 9, ..., 16$ (aprox. II).

In a second experiment, we have tested the performance of our algorithm when the matrix $\mathbf{W}$ obtained in the first experiment is used to compress other images. Figure 7 shows the original image "goldhill" and the recovered image obtained from the seven statistically independent components with largest energy. We can see that the visual quality is high. For this image, we have obtained a PSNR of 28.41 dB.

VI. Conclusions

In natural images, most of the important information is contained in the high-order relationships among the pixels. This property is not considered in PCA-based algorithms because only second-order statistics are used. In this paper, we have proposed an image compression algorithm which uses non-linear functions to extract the statistically independent components in the image instead of uncorrelated components. Simulation results show that the image representations based on statistically independent components are more adequate than representations obtained from decorrelated components.

Fig. 3. Experiment I: Original image "peppers"

Appendix

I. Separating Points of GEASI Algorithm

In section IV, we have determined that the points where the statistically independent components are extracted correspond to stationary points of the GEASI algorithm. In this appendix we will find the conditions which guarantee that these points are attractors of the algorithm. For simplicity reasons, this analysis is performed considering only two components. First we will write the expression (5) in

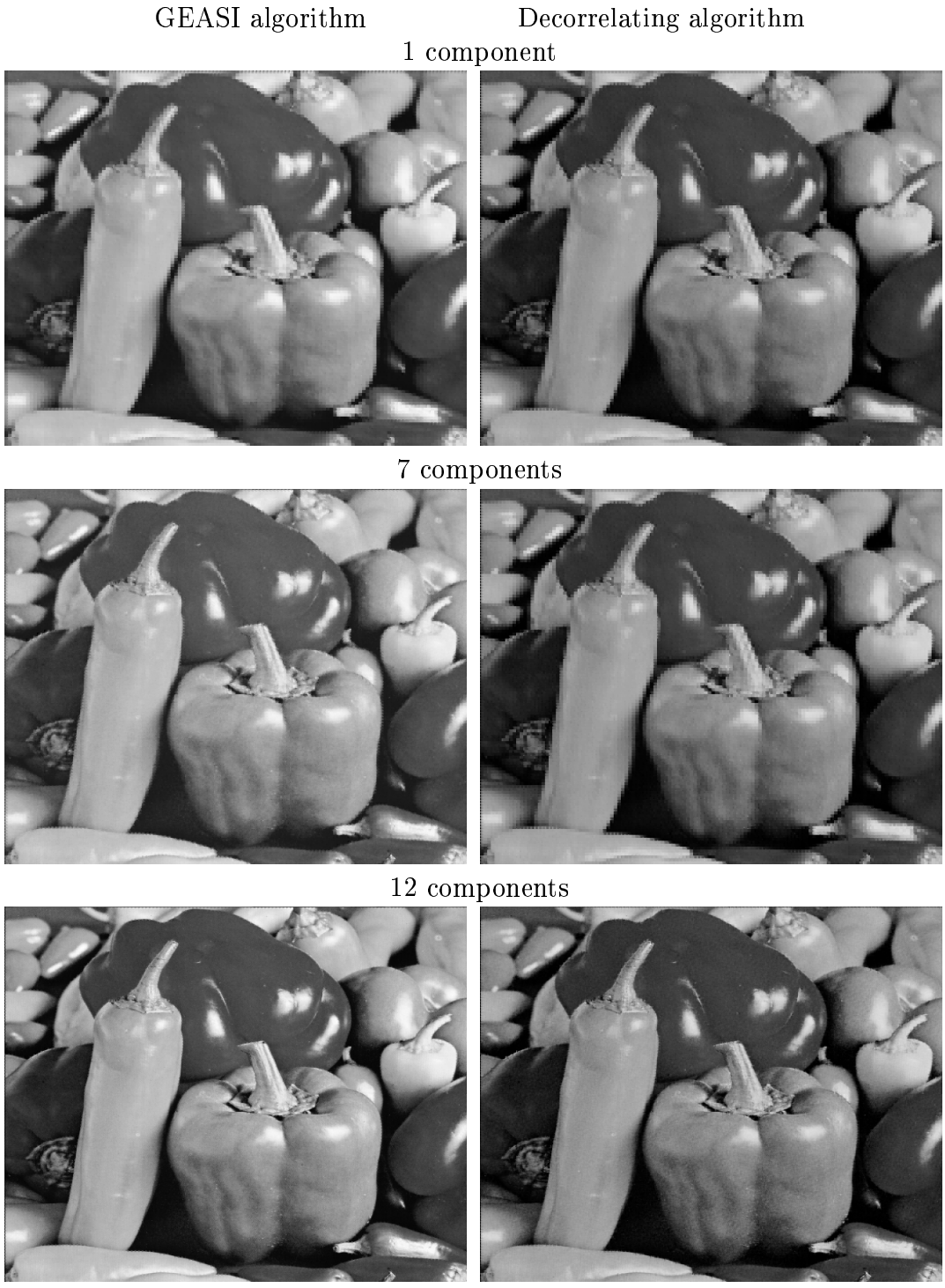

Fig. 4. Recovered images for several retained components using the GEASI algorithm (left) and the decorrelating algorithm (right).

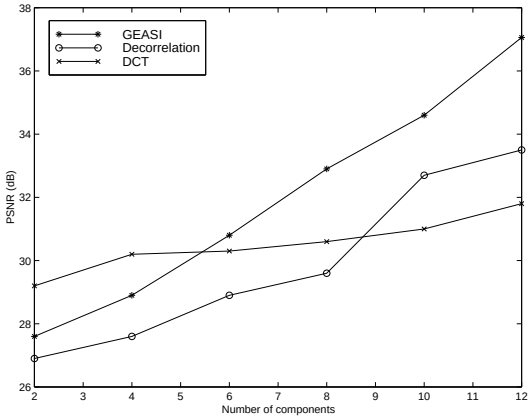

Fig. 5. Experiment I: PSNR obtained using the GEASI algorithm, the decorrelating algorithm and the DCT

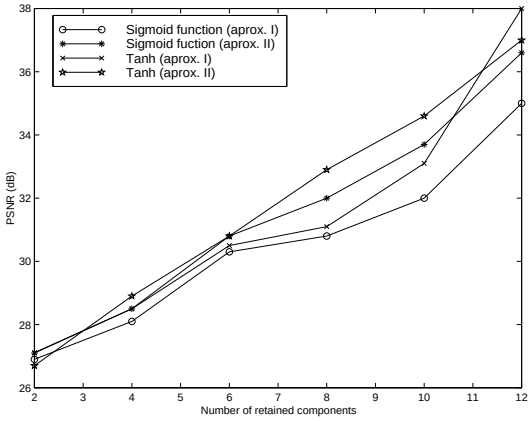

Fig. 6. Experiment I: PSNR obtained using several non-linear functions

Component	Energy	MSE
2	0.6667	158.7164
3	0.6677	159.1449
7	0.6842	160.9255
6	0.6865	160.7633
5	0.7336	157.7315
8	0.7424	158.9404
4	0.7946	155.8674
1	0.8101	154.5680
11	0.8174	155.5581
10	0.8386	155.7895
15	0.8536	144.5303
14	0.8766	146.5909
9	0.8772	145.2187
12	0.9207	144.5720
16	0.9653	139.0012
13	0.9713	140.2258

TABLE I

EXPERIMENT I: ENERGY AND MSE FOR EACH COMPONENT

(a)

(b)

Fig. 7. Experiment II. (a) Original image "goldhill" and (b) recovered image

terms of the mixing/separating matrix $\mathbf{G} = \mathbf{W}^T\mathbf{A}$

$$\begin{aligned}\frac{d\mathbf{G}^T}{dt} &= -\mathbf{G}^T\left(E[\mathbf{y}\mathbf{y}^T] - \mathbf{I}\right. \\ &+ \left. E[\mathbf{y}\mathbf{g}^T(\mathbf{y})]\mathcal{E} - \mathcal{E}E[\mathbf{g}(\mathbf{y})\mathbf{y}^T]\right) \quad (11)\end{aligned}$$

In the particular case of two components, we obtain

$$\begin{aligned}F_{11} = \frac{dg_{11}}{dt} &= -g_{11}(E[y_1^2] - 1) - g_{21}\left(E[y_2y_1]\right. \\ &+ \left.\epsilon_1 E[y_2 g(y_1)] - \epsilon_2 E[g(y_2)y_1]\right) \\ F_{12} = \frac{dg_{12}}{dt} &= -g_{12}(E[y_1^2] - 1) - g_{22}\left(E[y_2y_1]\right. \\ &+ \left.\epsilon_1 E[y_2 g(y_1)] - \epsilon_2 E[g(y_2)y_1]\right) \\ F_{21} = \frac{dg_{21}}{dt} &= -g_{11}\left(E[y_2y_1]\epsilon_2 E[y_1 g(y_2)]\right. \\ &- \left.\epsilon_1 E[g(y_1)y_2]\right) - g_{21}(E[y_2^2] - 1) \\ F_{22} = \frac{dg_{22}}{dt} &= -g_{12}\left(E[y_2y_1]\epsilon_2 E[y_1 g(y_2)]\right. \\ &- \left.\epsilon_1 E[g(y_1)y_2]\right) - g_{22}(E[y_2^2] - 1)\end{aligned}$$

Now, the derivatives of these equations respect to the coefficients in $\mathbf{G}$ are

$$\frac{\partial F_{ij}}{\partial g_{ij}} = -E[y_i^2] + 1 - 2g_{ij}E[y_i s_j]$$

$$
\begin{aligned}
&- \sum_{\substack{k=1 \\ k\neq i}}^{2} g_{kj}(E[y_k s_j] + \epsilon_j E[y_k g'(y_i) s_j] \\
&- \epsilon_k E[g(y_k) s_j]) \;\; i,j = 1,2 \\
\frac{\partial F_{ij}}{\partial g_{il}} &= -2 g_{ij} E[y_i s_l] - \sum_{\substack{k=1 \\ k\neq i}}^{2} g_{kj}(E[y_k s_l] \\
&+ \epsilon_i E[y_k g'(y_i) s_l] - \epsilon_k E[g(y_k) s_l]) \\
&\qquad i,j,l = 1,2, \;\; l \neq j \\
\frac{\partial F_{ij}}{\partial g_{lj}} &= -E[y_l y_i] - \epsilon_i E[y_l g(y_i)] + \epsilon_l E[g(y_l) y_i] \\
&- g_{lj}(E[s_j y_i] + \epsilon_i E[s_j g(y_i)] \\
&- \epsilon_i E[g'(y_l) s_j y_i]) \;\; i,j,l = 1,2, \;\; l \neq i \\
\frac{\partial F_{ij}}{\partial g_{ml}} &= -g_{mj}(E[s_l y_i] + E[s_l g(y_i)] \\
&- E[g'(y_m) s_l y_i]) \;\; i,j,m,l = 1,2, \;\; m \neq i, \;\; l \neq j
\end{aligned}
$$

Most of the second derivatives vanish in the point where each output extract a single component, $\mathbf{G} = \mathbf{I}$. In fact, the only non-zero derivatives are

$$
\begin{aligned}
\frac{\partial F_{11}}{\partial g_{11}} &= -2 \\
\frac{\partial F_{12}}{\partial g_{12}} &= -1 - \epsilon_1 E[g'(s_1)] + \epsilon_2 E[g(s_2) s_2] \\
\frac{\partial F_{12}}{\partial g_{21}} &= -1 - \epsilon_1 E[g(s_1) s_1] + \epsilon_2 E[g'(s_2)] \\
\frac{\partial F_{21}}{\partial g_{12}} &= -1 - \epsilon_2 E[g(s_2) s_2] + \epsilon_1 E[g'(s_1)]) \\
\frac{\partial F_{21}}{\partial g_{21}} &= -1 - \epsilon_2 E[g'(s_2)] + \epsilon_1 E[g(s_1) s_1] \\
\frac{\partial F_{22}}{\partial g_{22}} &= -2 \qquad (12)
\end{aligned}
$$

Finally, the matrix formed by the second derivatives is given by

$$
\mathbf{H} = \begin{bmatrix}
\frac{\partial F_{11}}{\partial g_{11}} & \frac{\partial F_{11}}{\partial g_{12}} & \frac{\partial F_{11}}{\partial g_{21}} & \frac{\partial F_{11}}{\partial g_{22}} \\
\frac{\partial F_{12}}{\partial g_{11}} & \frac{\partial F_{12}}{\partial g_{12}} & \frac{\partial F_{12}}{\partial g_{21}} & \frac{\partial F_{12}}{\partial g_{22}} \\
\frac{\partial F_{21}}{\partial g_{11}} & \frac{\partial F_{21}}{\partial g_{12}} & \frac{\partial F_{21}}{\partial g_{21}} & \frac{\partial F_{21}}{\partial g_{22}} \\
\frac{\partial F_{22}}{\partial g_{11}} & \frac{\partial F_{22}}{\partial g_{12}} & \frac{\partial F_{22}}{\partial g_{21}} & \frac{\partial F_{22}}{\partial g_{22}}
\end{bmatrix} =
$$

$$
\begin{bmatrix}
-2 & 0 & 0 & 0 \\
0 & \begin{array}{c} -1 - \epsilon_1 E[g'(s_1)] + \\ + \epsilon_2 E[g(s_2) s_2] \end{array} & \begin{array}{c} -1 - \epsilon_1 E[g(s_1) s_1] \\ \epsilon_2 E[g'(s_2)] \end{array} & 0 \\
0 & \begin{array}{c} -1 - \epsilon_2 E[g(s_2) s_2] \\ + \epsilon_1 E[g'(s_1)] \end{array} & \begin{array}{c} -1 - \epsilon_2 E[g'(s_2)] \\ \epsilon_1 E[g(s_1) s_1] \end{array} & 0 \\
0 & 0 & 0 & -2
\end{bmatrix}
$$

The matrix $\mathbf{H}$ has four eigenvalues: $e_1 = e_2 = e_3 = -2$ and $e_4 = \mathcal{C}_{s_1} + \mathcal{C}_{s_2}$ where $\mathcal{C}_{s_i} = \epsilon_i(E[s_i g(s_i)] - E[g'(s_i)])$. It is easy to find that the following condition guarantees that the eigenvalues have negative real part

$$
\begin{aligned}
\mathcal{C}_{s_1} + \mathcal{C}_{s_2} < 0 \Rightarrow \\
\epsilon_1(E[s_1 g(s_1)] - E[g'(s_1)]) + \\
\epsilon_2(E[s_2 g(s_2)] - E[g'(s_2)]) < 0 \qquad (13)
\end{aligned}
$$

As a consequence, in order to guarantee that the desired solutions are attractors of the GEASI algorithm, the non-linear function $g(.)$ must satisfy the condition (13). In the general case of N statistically independent components, the non-linear function $g(x)$ must guarantee

$$
\mathcal{C}_{s_i} + \mathcal{C}_{s_j} < 0 \;\; i,j = 1,...,N, \;\; i \neq j \qquad (14)
$$

where $\mathcal{C}_{s_i} = \epsilon_i(E[s_i g(s_i)] - E[g'(s_i)])$.

References

[1] S. Haykin, *Neural Networks. A Comprehensive Foundation*, Macmillan College Publishing Company, New York, 1994.

[2] T-W. Lee, M. Girolami, T. J. Sejnowski, "Independent Component Analysis Using an Extended Infomax Algorithm for Mixed Sub-Gaussian and Super-Gaussian Sources", *Neural Computation*, in press.

[3] J. Karhunen, A. Hyvarinen, R. Vigario, J. Hurri, E. Oja, "Applications of Neural Blind Separation to Signal and Image Processing", *proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 131-134, Munich, Germany, 1997.

[4] S. Bartlett, M. Lades, T. J. Sejnowski, "Independent component representations for face recognition", *proc. of the SPIE*, vol 3299, pp. 528-539, California, 1998

[5] J. Karhunen, "Neural Approaches to Independent Component Analysis and Source Separation", *proc. European Symposium on Artificial Neural Networks*, pp. 249-266, Bruges, Belgium, April 1996.

[6] J. F. Cardoso, "Blind Signal Separation: Statistical Principles", *Proceedings of IEEE*, vol. 86, no. 10, pp. 2009-2025, October 1998.

[7] P. Comon, "Independent Component Analysis, A New Concept?", *Signal Processing*, Vol. 36, 1994, pp. 287–314.

[8] X. Cao and R. Wen-Liu, "General Approach to Blind Source Separation", *IEEE Transactions on Signal Processing*, Vol. 44, No. 4, 1996, pp. 562–571.

[9] T-W. Lee, *Independent Component Analysis: Theory and Applications*, Kluwer Academic Publishers, Boston, 1998.

[10] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall Inc., New Jersey, 1989.

[11] J. F. Cardoso and B. Laheld, "Equivariant Adaptive Source Separation", IEEE Trans. on Signal Processing, vol, 44, no. 12, pp. 3017-3030, December 1996.

[12] M. W. Hirsch and S. Smale, *Differential Equations, Dynamical Systems, and Linear Algebra*, Academic Press, New York, 1974.

[13] J. G. Blinn, "What's the Deal with the DCT?", *IEEE Computer Graphics and Applications*, vol. 13, pp. 78-83, July 1993.